Shamir’s Secret Sharing

Shamir’s Secret Sharing is one of the first secret sharing schemes in cryptography, based on polynomial interpolation over finite fields.

Polynomial interpolation consists in defining a polynomial of the lowest degree possible that passes through all points of a dataset. As a simple example, the polynomial f(x)=x+1 would interpolate a dataset with (-1,0), (1,2), (3,4), etc… (grafic)

In order to secure a secret using Shamir’s Secret Sharing, the secret itself is split into shares (the points of the dataset), which will be used to reconstruct the original secret.

To unlock the secret, a minimum number of shares is required, called the threshold. Shamir’s Secret Sharing achieves perfect secrecy because an attacker that has discovered a number of shares lower than the threshold has not learned any information about the secret.

Shamir’s Secret Sharing is based on the Lagrange Interpolation Problem, specifically that k points are enough to define a polynomial of degree k - 1 (2 points are enough to define a line, 3 points are enough to define a parabola, …

This way, we can define our secret as an element of a finite field, and then construct a polynomial with our secret as the constant element and k - 1 other random elements.



Every participant is given a different point in this polynomial. With k different points, the original polynomial can be rebuilt using interpolation and the original secret can be found.

Now we will see an example that does not use finite fields, and we’ll discuss why it doesn’t provide complete secrecy. Then, we will see how finite fields fix this problem.

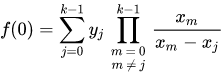
First we must choose our secret S = 1789, the number of total shares n = 6 and how many shares are needed to unlock the secret k= 3. We obtain k - 1 numbers, a1 = 1643 and a2 = 805. We build a polynomial of k - 1 degree using these numbers, and S as a0, the constant.

f(x) = 1789 + 1643x + 805x^2

Now we can give each participant their points, starting at 1 because f(0) = S.

(1, 4237), (2, 8295), …

The original polynomial and thus the secret can be reconstructed using any subset of three points of the shares, using Lagrange Basis Interpolation.



This formula calculates the scaled basis polynomials of each point (polynomials that pass through the point and are 0 where the x is of another point) and then sums them to obtain the interpolation polynomial. The constant in the polynomial is the secret S.

Now we will see why not using finite fields does not provide perfect secrecy. Let’s assume that a spy has obtained two shares. In theory, since they do not have 3 shares they shouldn’t have learnt anything about the secret, but this is false. The spy can combine the information in the shares they have obtained and the public info.

Assuming the spy has the first two points (1, 4237), (2, 8295) they can obtain extra info with this procedure:

Filling f(x) with the value of k: f(x) = S + a1x + a2x^2

Filling the formula with the values in the points: f1 -> 4237 = S + a1·1 + a2·1^2 = S + a1 + a2

f2 -> 8295 = S + a1·2 + a2·2^2 = S + a1·2 + a2·4

Calculates f2 - f1 to obtain f3 = (8295 - 4237) = (S - S) + (2a1 - a1) + (4a2 - a2) -> 4058 = a1 + 3a2 and rewrites it as a1 = 4058 - 3a2

Since we know that a2 is a natural number, the spy starts replacing a2 by 0, 1, 2, … until a1 is negative, because ax is a natural number.

The spy now knows all possible values for a2.

Replaces a1 by f3 in f1 to obtain 4237 = S + (4058 - 3a2) + a2 -> S = 179 + 2a2.

Now the spy can replace a2 by all the possible values and will find a range for the secret way smaller than the infinite number of natural numbers.

This is solved using finite fields, because this attack exploits the fact that the points must lie on a smooth curve. If instead we represent our polynomial over a finite field instead of the natural numbers, we see that it becomes very disorganized. To cover our secret, we just need a small change. We need a prime p larger than S and all ax’s, and now our points will be (x, f(x) mod p) instead of (x, f(x)). Now the spy cannot learn any information with less than k shares because… explicacio

Explicar les properties